Proceedings of the Fifth IIEEJ International Workshop
on Image Electronics and Visual Computing 2017
Da Nang, Vietnam, February 28 - March 3, 2017

SPLIGHT: LIGHTING FOR SPLAT BASED RENDERING TOWARDS TEMPORAL COHERENCE

Muhammad Arief†
†Tokyo University of Technology

Hideki Todo†
†Tokyo University of Technology

Koji Mikami†
†Tokyo University of Technology

Kunio Kondo†
†Tokyo University of Technology

ABSTRACT

Existing brush splat method can produce painterly look animation with temporal coherence but limited to the manual key framing process whereas lighting effects need more dynamic interaction. Another type of stroke style rendering is example-based method that can automatically transfer reference styles from 3D lighting results. However, its frame by frame process may cause the lack of temporal coherence in generated animation. In this study, we present a rendering framework to animate lighting effects while preserving temporal coherence in brush stroke styles. Our method constructed as combining 3D textured splat and dynamic texture projection. Our real-time system also gives controls over key visual elements of brush strokes such as shape, distribution, and color.

1. INTRODUCTION

Non-photorealistic styles applied to the artworks can create the imaginary impression to the viewers. Such an artistic style is usually found in the 2D environment since it gives the flexibility for the artist to draw expressively without considering many parameters. The expressive style found in brush stroke requires a lot of attention when it comes to animate because it contains many decorative features for keeping the temporal coherence either in 2D and 3D environment. Among them, lighting effects can be considered as a challenging target since light is the most moving part of the animation.

In this study, we present an approach which contributes for creating real-time 3D lighting framework for stroke-based rendering. Our system allows the artist to quickly create lighting effects in brush stroke style with the key brush stroke features such as size, position, and color. These features are dynamically updated frame by frame, while maintaining the temporal coherence in animation results.

Figure 1 shows a typical hand-painted brush animation example designed by an artist. To animate lighting effects from frame (a) to (d) with brush style appearance, the artist needs to carefully design the lighting transition of all brush elements in each frame manually. This situation is not effective for the tight schedule in animation production.

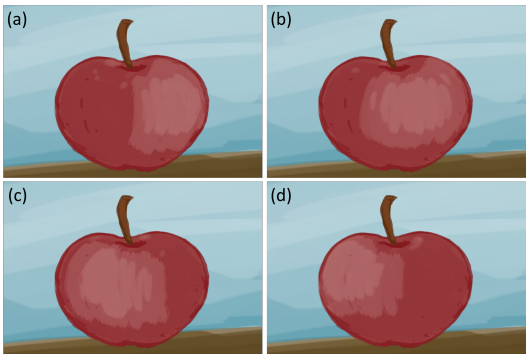

Figure 1: Brush animation example hand-painted by an artist. Liselotte Heimdahl ©

2. PREVIOUS WORKS

Many types of research have been conducted to create brush stroke style animation while few researches pay attention to lighting effects in those style animation.

Various 3D software such as Autodesk Maya ® and 3ds-max ® have capabilities of creating animation of brush stroke styles by using simple texture based methods. While this approach can produce a variety of brush stroke appearance but few interactions are available for designing the animation of the target brush stroke style.

Mackiewicz and Melendez demonstrated high quality hand-painted animation [1] which is designed by painting each frame one by one. While the manual design process is suitable for reproducing the original painting style of Vincent van Gogh, this strategy takes a lot of time and requires many skilled painters.

In 3D painting *Overcoat* [2] can produce painterly rendering styles for a 3D object by placing textured splats over the model. This technique is extended by Basset et al. [3]

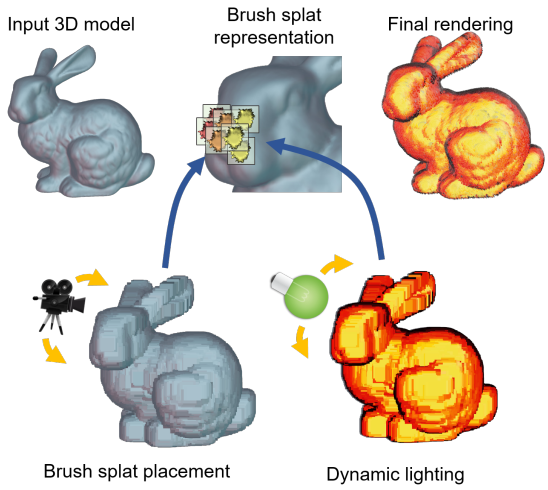

Figure 2: Method overview. Based on the brush splat representation, our system dynamically updates brush stroke placement and colors for the final rendering result.

which enables keyframe interactions for animating the textured splats.

Illumination context style transfer is proposed in *Stylit* [4] which operates style trasnfer process frame-by-frame separately, The transfer method is based on *Image Analogies* [5] technique which optimize correspondence between style and synthesized image by searching style references from an image example. However, its frame by frame process may lack the temporal coherence.

Todo et al. [6] extended the *lit-sphere* technique [7] by incorporating normal coordinates in light space view to provide lighting interaction for original normal-to-color texture lookup process. The advantage of dynamic lit-sphere shading is direct appearance design based on the 2D textures but still difficult to animate brush strokes.

3. METHOD

Figure 2 outlines the key idea of achieving the temporal coherence on the proposed method. Our stroke rendering framework is based on dynamic brush texture splats whose parameters are updated according to the light and camera. With a given input 3D model, our system first distributes splats for brush stroke placement (view-dependent). Next, each splat is colored by dynamic lighting process as described in Section 3.2 (light-dependent). Finally, we obtain the appearance of a stroke style by overlaying brush splats with a specified brush shape. The reason of choosing the splat representation is that it is commonly used in 3D environment and it is easier to animate the stoke appearance by changing the brush stroke parameters.

For brush stroke parameters, we refer the brush stroke analysis [8] which describes the key elements of brush stroke characteristics: distribution, shape, and orientation. In this study, we mainly focus on distribution and shape to incorporate these key elements in our brush splat representation. For the distribution, we control the brush splat placement by changing the target density over the 3D model. For the shape, our system allows the artist to use a brush stroke texture which clips the splat are like a stamp. For the orientation, our current system simply updates each splat to face toward the camera, while 2D rotation of brush splats may improve the visual quality. In addition to these spatial elements, we also use a brush color as a common visual element. In our rendering framework, each splat color is updated through dynamic lighting.

3.1. BRUSH TEXTURE SPLAT

With a given 3D model, our system generates 3D splats as a basis for brush stroke texture placement. The brush stroke placement process is initiated by distributing point cloud on the 3D space. Our system randomly samples the 3D object triangles and places the splat in the center of the sampled triangle. Each splat is placed on a sample of point cloud with a squared area for drawing.

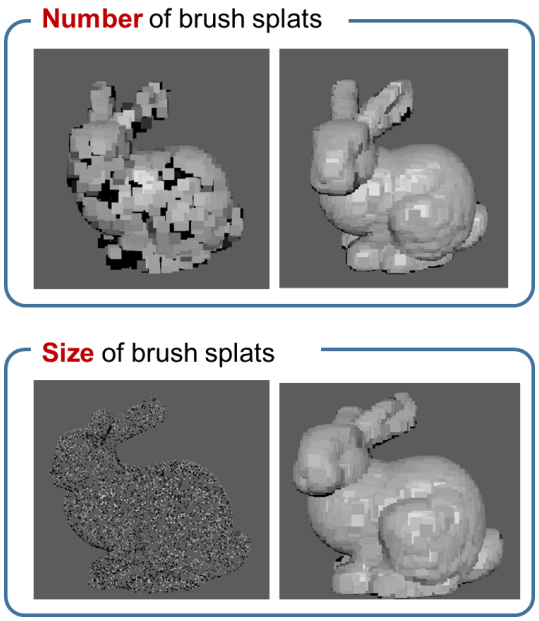

Figure 3: User control for brush splat placement. The user can adjust the number and size of brush splats for changing the brush stroke distribution.

Figure 3 illustrates the user control for brush splat placement. The user can set the number of the point cloud for the 3D model within the number of triangles available on the

3D object. The user can also specify a uniform size parameter which affects the size of all splats at once.

3.2. DYNAMIC LIGHTING

Figure 4 shows the overview of the dynamic lighting process in our rendering framework. This process is used for updating each splat color according to the lighting. To make a stylized shading appearance, we use the lit-sphere extension [6] that can transfer the designed shading reference onto the 3D model.

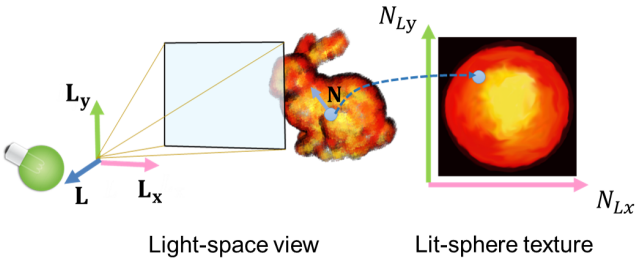

Figure 4: Dynamic lighting overview. Surface normal $\mathbf{N}$ is seen from the light space view to obtain light space coordinates N_{Lx} and N_{Ly}. These coordinates are used for referencing the lit-sphere texture. This method incorporate surface normal vector from each splat.

In the lit-sphere shading, first surface normal $\mathbf{N}$ is transformed into the light space view as:

$$\mathbf{N}_L = (N_{Lx}, N_{Ly}, N_{Lz}) \tag{1}$$

where $\mathbf{N}_L$ is the normal coordinates in the light space view which is spanned by $\mathbf{Lx}$, $\mathbf{Ly}$, and $\mathbf{L}$. Each coordinate can be computed as:

$$N_{Lx} = \mathbf{N} \cdot \mathbf{Lx}, \quad N_{Ly} = \mathbf{N} \cdot \mathbf{Ly}, \quad N_{Lz} = \mathbf{N} \cdot \mathbf{L} \tag{2}$$

where N_{Lx}, N_{Ly}, and N_{Lz} are defined as a simple dot product form of the surface normal $\mathbf{N}$ and light space coordinate axes $\mathbf{Lx}$, $\mathbf{Ly}$, and $\mathbf{L}$. Finally, we can obtain the shading color by 2D texture lookup process with the computed N_{Lx} (x coordinate) and N_{Ly} (y coordinate).

Based on the dynamic lighting process described in the above, we apply the lit-sphere shading to each splat as shown in Figure 5. We begin by computing the lit-sphere shading for the target 3D model. The shading result is used for determining the splat colors by referencing the color of the original sampled point on the surface. Finally, we obtain a brush stroke appearance by using an additional brush stroke texture which is used for creating a brush shape by alpha clipping.

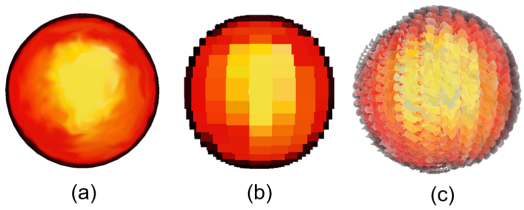

Figure 5: Brush splat shading overview. (a) Lit-sphere shading result on the 3D model. (b) Splats shaded by sampling colors of the lit-sphere shading. (c) Final brush splat appearance clipped with brush stroke shapes.

4. RESULT AND DISCUSSION

To build an artist-friendly system based on the proposed method, we implemented our system as a CgFx Shader (a GPU shading process) in a standard 3D software Maya ®. With NVIDIA Geforce GTX 980M, we can achieve real-time performance for the preview.

Figure 6 and the supplemental video show final animation results designed with our system. We designed dynamic lighting effects with a single point light source. The brush splat distribution is adjusted by the number of the splats with a suitable size setting to fill the gaps between the splats.

The shape element within the brush texture on each splat can be easily changed by choosing the 2D image brush shape reference which are effective to set the stroke style appearance immediately. Including the color element obtained from lit-sphere texture, our real-time rendering framework provides a way to quickly test the variety of stroke styles by changing the key brush stroke elements.

While the splat placement is static on the target 3D model, we can observe that the transition of brush stroke appearance. In making the transition of the shading, the brush splat orientation is basically affected by the camera. The remained brush colors are determined by the light setting through our dynamic lighting process.

Our method manages stroke elements as view and light dependent parameters which will be coherently updated by both the camera and light. On the other hand, *Stylit* [4] brushstroke parameters such as distribution, shape, and orientation are drastically changed frame-by-frame.

As for the temporal coherence, colors are smoothly changed via lit-sphere technique shading which is commonly used for dynamic lighting of a 3D surface. The main difference between a splat and 3D surface is that each splat is layered with a finite squared area. which may cause popping artifacts. Such artifacts are less noticeable with a small size of brush splat as shown in the supplemental video.

Figure 7 compares our rendering results by changing distribution parameters in brush splats. To make these results,

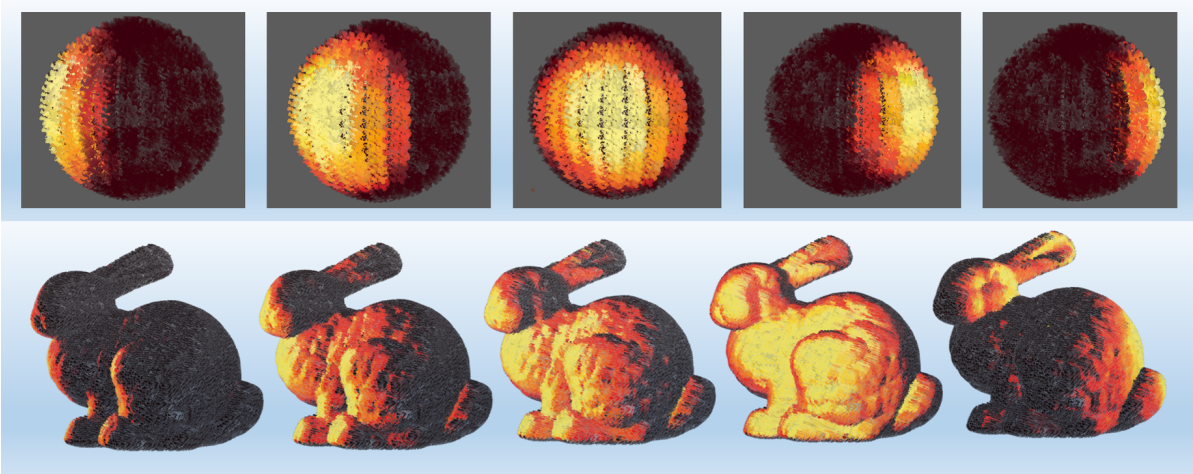

Figure 6: Animated sequence of rendering results through dynamic lighting.

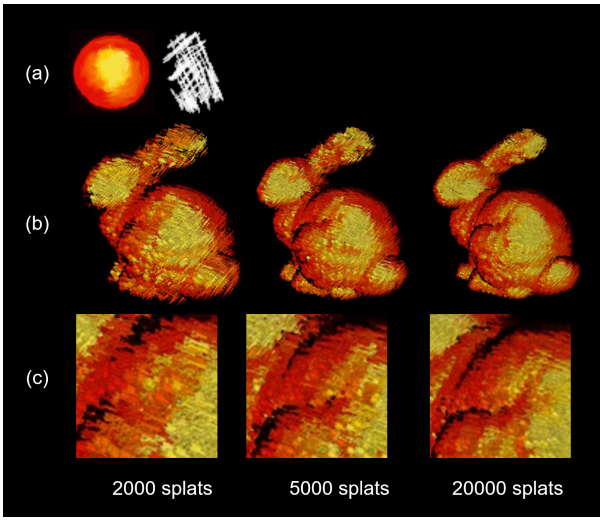

Figure 7: Rendering effects by changing the number of splats. (a) Set of lit-sphere texture and brush stroke texture used for rendering the scene. (b) Rendering results with different numbers of brush splats. (c) Close-up views of the rendering results.

we use the same lit-sphere texture and brush texture in the figure. Depending on the number of splats, we adjust the size parameter to fill the gap of the splat distribution. By changing the density of splats, we can obtain the different appearance of the final rendering styles.

As shown in Figure 8, our system can easily change the appearance of the object by changing the lit-sphere texture. To compare the lit-sphere shading effects, we use the same brush stroke texture for all examples. We tested the different shading appearances: soft quantization effects, typical constant tones, and photorealistic metallic reflectance. We observed that even photorealistic styles can be changed into a brush stroke appearance. Our system provides a flexibility to choose a base stroke shading style with the lit-sphere texture.

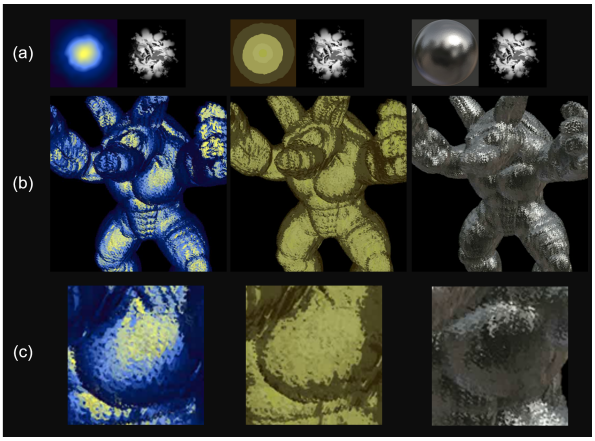

Figure 8: Rendering effects by changing lit-sphere textures. (a) Lit-sphere textures and a brush stroke texture used for rendering the scene. (b) Rendering results with different lit-sphere textures. (c) Close-up views of the rendering results.

5. CONCLUSION AND FUTURE WORK

In this study, we propose a stroke rendering framework based on brush splats whose parameters are dynamically updated according to the light and camera. While we can obtain dynamic lighting results with stroke rendering styles,

there are many functions that need to be developed in order to give the artist more control over the result.

First, brush stroke orientation is static in our current implementation. We need to evaluate how the attribute affects final animation. In actual brush strokes in existing painted artworks, the stroke orientations are changed along the surface structure or the illumination. We would like to incorporate such inherent elements with our system to control the orientation.

Second, our brush splat representation is limited to distributing small elements. Therefore, a long sequence of stroke path is difficult to design using our system. Extending our rendering framework to stroke path is an essential future direction to design a more variety of stroke styles in generated animations.

While these elements are more complicated than the animated elements in this study, we hope that our initial experiment will give a key insight for achieving a lighting behavior in a suitable temporal coherent manner.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 15K00508 and LPDP (indonesia endowment fund for education)

REFERENCES

[1] Lukasz Mackiewicz and Francho Melendez. 2016. Loving vincent: guiding painters through 64.000 frames. In ACM SIGGRAPH 2016 Talks (SIGGRAPH '16). ACM, New York, NY, USA, Article 6, 2 pages.

[2] Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert W. Sumner. 2011. OverCoat: an implicit canvas for 3D painting. ACM Trans. Graph. 30, 4, Article 28 (July 2011), 10 pages.

[3] Katie Bassett, Ilya Baran, Johannes Schmid, Markus Gross, and Robert W. Sumner. 2013. Authoring and animating painterly characters. ACM Trans. Graph. 32, 5, Article 156 (October 2013), 12 pages.

[4] Jakub Fier, Ondej Jamrika, Michal Luk, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Skora. 2016. StyLit: illumination-guided example-based stylization of 3D renderings. ACM Trans. Graph. 35, 4, Article 92 (July 2016), 11 pages.

[5] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. 2001. Image analogies. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01). ACM, New York, NY, USA, 327-340.

[6] Hideki Todo, Ken Anjyo, and Shun'Ichi Yokoyama. 2013. Lit-Sphere extension for artistic rendering. Vis. Comput. 29, 6-8 (June 2013), 473-480.

[7] Peter-Pike J. Sloan, William Martin, Amy Gooch, and Bruce Gooch. 2001. The lit sphere: a model for capturing NPR shading from art. In Proceedings of Graphics Interface 2001 (GI '01). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 143-150.

[8] Igor E. Berezhnoy, Eric O. Postma, and H. Jaap Herik. 2009. Automatic extraction of brushstroke orientation from paintings. Mach. Vision Appl. 20, 1 (January 2009), 1-9.